LEC 9

CSE 122

Nested Collections

Questions during Class?

Raise hand or send here

sli.do #cse122

Slido vote & chat with neighbors:

The weather looks fabulous! What are your weekend plans?

Music: 122 26sp Lecture Jams 🎧

Instructor: Elba Garza

TAs:

David	Caleb	Cole	Yang
William	Neha	Blake R.	Cady
Dani	Wesley	Carson	Diya
Rohan	Isis	Sushma	
Andrew	Colin	Connor	
Ava	Naomi	Mahima	
Shreyank	Hanna	Nicolae	
Nicole	Blake P.	Ivory	

Agenda (1/4)

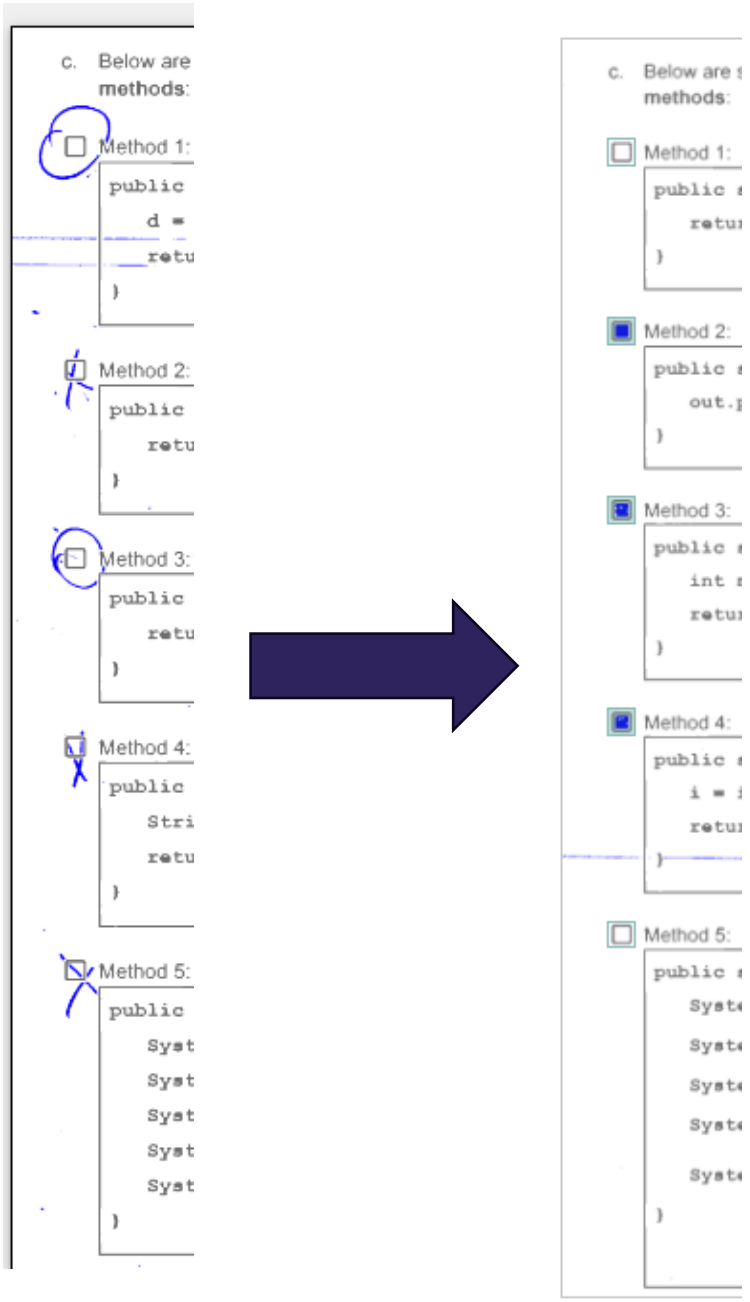
- Announcements
- Review: Maps & mostFrequentStart
- Recap: Nested Collections
- Practice: Social Network

Announcements

- Programming Assignment 2 (P2) releasing today!
  - Seriously, start early! This assignment is much more involved…
  - Due May 12th by 11:59pm PT
- Quiz 1 on May 7th in your registered Quiz Section
  - Topics: (Reference Semantics), Stacks and Queues, Sets, Maps
  - Practice Quiz 1 available some time this weekend; solutions on Tuesday
- Quiz 0 grades to be released this weekend!
- Resubmission Cycle 2 (R2) form out, due May 5th by 11:59pm PT
  - Available assignments: **C0**, P0, C1
  - Reminder: to use a resubmission cycle you need to
    - (1) submit your work (big blue "Submit" button on Ed)
    - **AND** (2) fill out the resubmission form (linked from Ed + course calendar)

An aside for quizzes… (1/2)

Please be legible and clear on your written answers 😬



An aside for quizzes… (2/2)

Please write your name legibly!

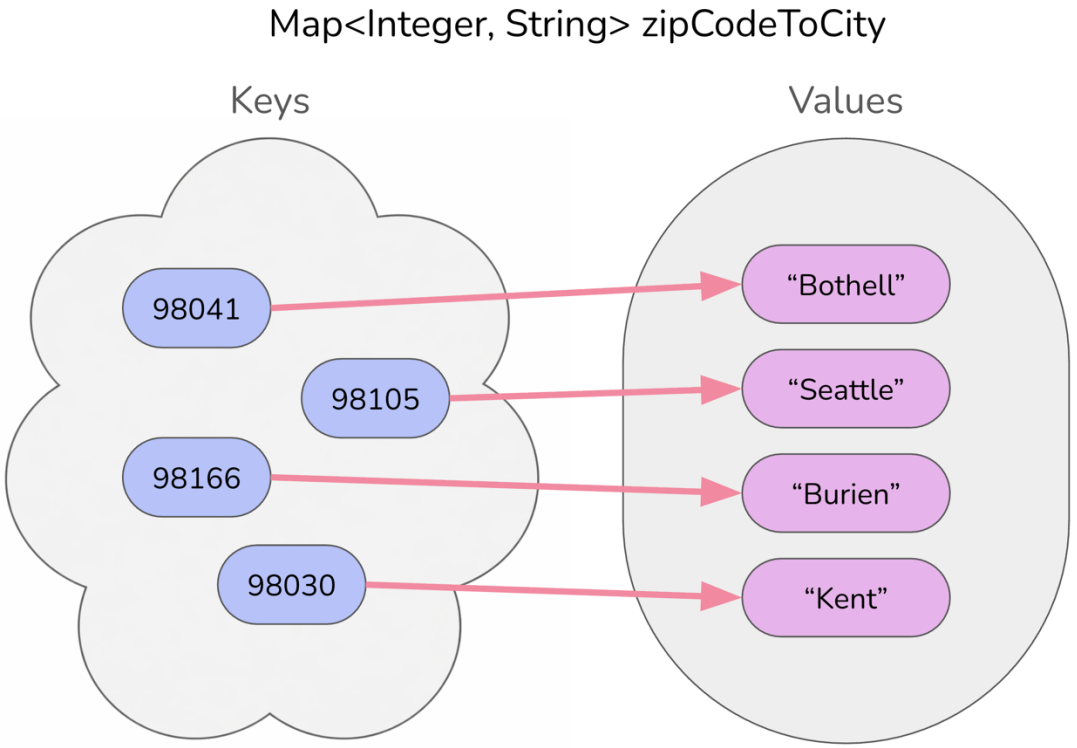
Name of Student: CSE 122

Agenda (2/4)

- Announcements
- **Review: Maps & mostFrequentStart**
- Recap: Nested Collections
- Practice: Social Network

Map ADT

- Data structure to map keys to values
  - Keys can be any* type; Keys must be unique
  - Values can be any type
- Example: Mapping ticker to stock price in P0
- Operations
  - `put(key, value)`: Associate key to value
    - Overwrites duplicate keys
  - `get(key)`: Get value for key
  - `remove(key)`: Remove key/value pair



mostFrequentStart (2/2)

Write a method called `mostFrequentStart` that takes a Set of words and does the following steps:

- Organizes words into "word families" based on which letter they start with
- Selects the largest "word family" as defined as the family with the most words in it
- Returns the starting letter of the largest word family (and *should update the Set of words to only have words from the selected family*).

mostFrequentStart (2/2)

For example, if the `Set` words stored the values:

```
["hello", "goodbye", "library", "literary", "little", "repel"]
```

The word families produced would be:

```
'h' -> 1 word ("hello")
'g' -> 1 word ("goodbye")
'l' -> 3 words ("library", "literary", "little")
'r' -> 1 word ("repel")
```

Since `'l'` has the largest word family, we **modify the `Set`** to only contain `Strings` starting with `'l'` and finally **return `'l'`**.

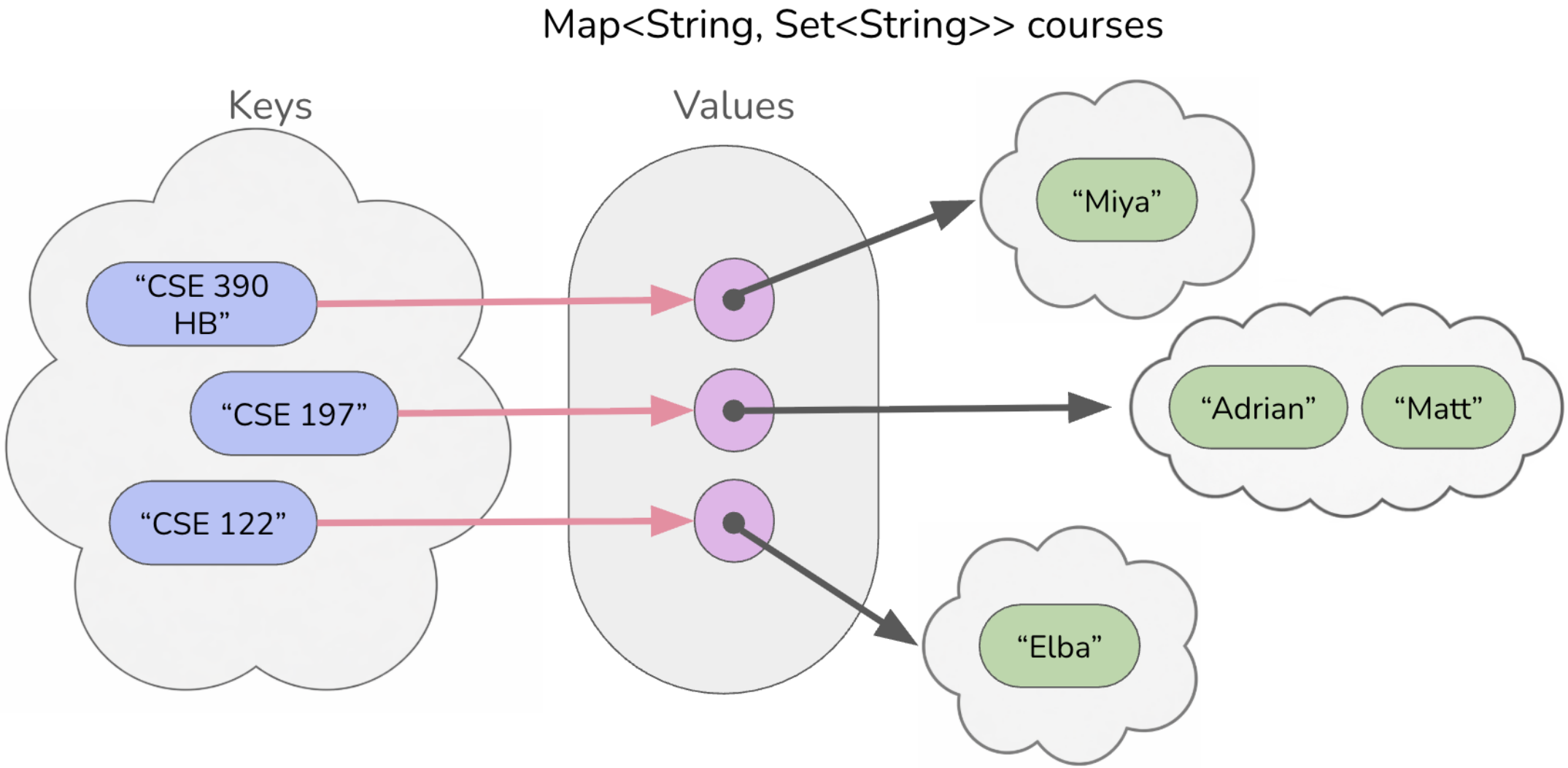
Agenda (3/4)

- Announcements
- Review: Maps & mostFrequentStart
- **Recap: Nested Collections**
- Practice: Social Network

Nested Collections (1/2)

- The values inside a Map can be any type, including data structures
- Common examples:
  - Mapping: Section ➔ **Set of students** in that section
  - Mapping: Recipe ➔ **Set of ingredients** in that recipe
    (Or even Map<String, Map<String, Double>> for units!)

Nested Collections (2/2)



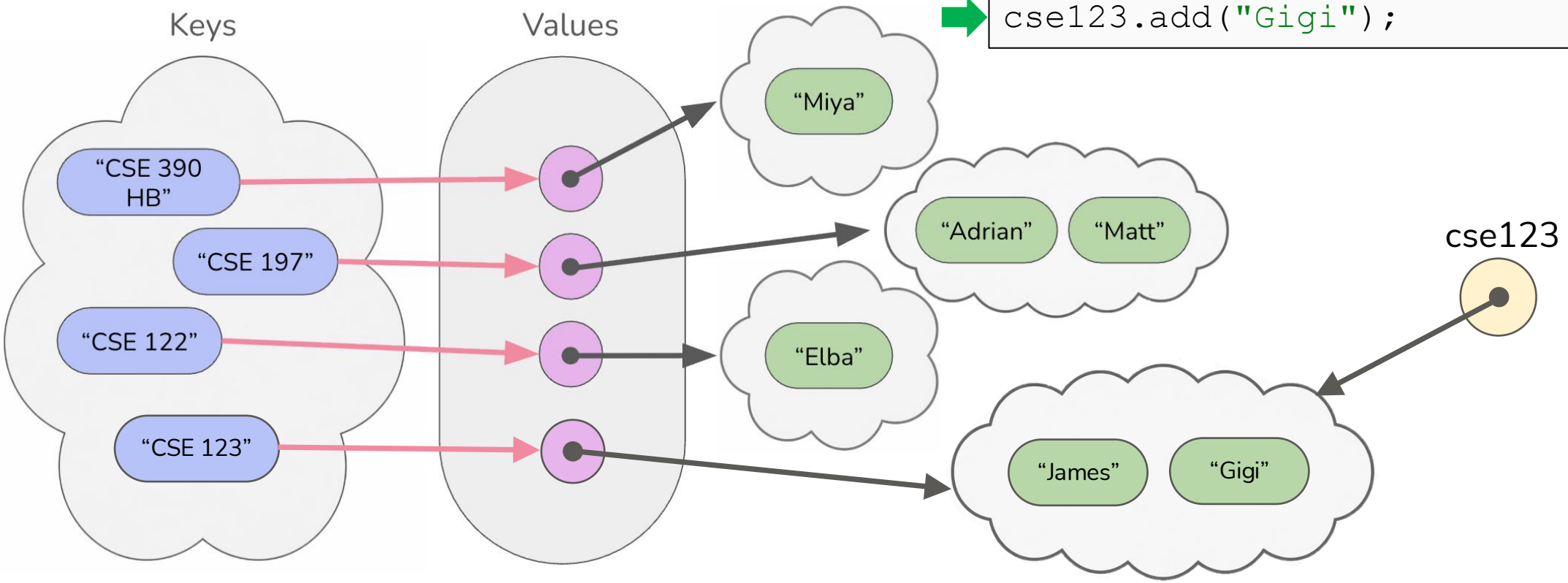
Updating Nested Collections

The "value" inside the Map is a reference to the data structure!

- Think carefully about number of references to a particular object

```
courses.put("CSE 123",
        new HashSet<String>());
courses.get("CSE 123").add("James");

Set<String> cse123 =
        courses.get("CSE 123");
cse123.add("Gigi");
```



Practice : Think

sli.do #cse122

Suppose map had the following items. What would its items be after running this code?

```
map: {"KeyA"=[1, 2], "KeyB"=[3], "KeyC"=[4, 5, 6]}
```

```
Set<Integer> nums = map.get("KeyA");
nums.add(7);
map.put("KeyB", nums);
map.get("KeyA").add(8);
map.get("KeyB").add(9);
```

A. `{"KeyA"=[1, 2], "KeyB"=[1, 2, 7], "KeyC"=[4, 5, 6]}`

B. `{"KeyA"=[1, 2, 8], "KeyB"=[1, 2, 7, 9], "KeyC"=[4, 5, 6]}`

C. `{"KeyA"=[1, 2, 7, 8], "KeyB"=[1, 2, 7, 9], "KeyC"=[4, 5, 6]}`

D. `{"KeyA"=[1, 2, 7, 8, 9], "KeyB"=[1, 2, 7, 8, 9], "KeyC"=[4, 5, 6]}`

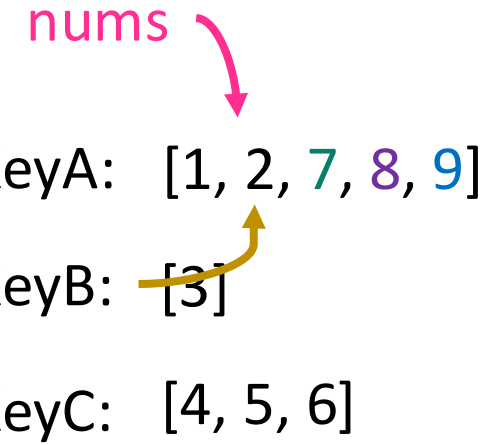
Practice : Pair

sli.do #cse122

Suppose map had the following items. What would its items be after running this code?

```
map: {"KeyA"=[1, 2], "KeyB"=[3], "KeyC"=[4, 5, 6]}
```

```
Set<Integer> nums = map.get("KeyA");
nums.add(7);
map.put("KeyB", nums);
map.get("KeyA").add(8);
map.get("KeyB").add(9);
```



A. `{"KeyA"=[1, 2], "KeyB"=[1, 2, 7], "KeyC"=[4, 5, 6]}`

B. `{"KeyA"=[1, 2, 8], "KeyB"=[1, 2, 7, 9], "KeyC"=[4, 5, 6]}`

C. `{"KeyA"=[1, 2, 7, 8], "KeyB"=[1, 2, 7, 9], "KeyC"=[4, 5, 6]}`

D. `{"KeyA"=[1, 2, 7, 8, 9], "KeyB"=[1, 2, 7, 8, 9], "KeyC"=[4, 5, 6]}`

Agenda (4/4)

- Announcements
- Review: Maps & mostFrequentStart
- Recap: Nested Collections
- **Practice: Social Network**